High Level Architecture (HLA) Performance Framework

Judith S. Dahmann, PhD
Defense Modeling and Simulation Office, 1901 N. Beauregard St, Ste 504, Alexandria, VA 22311, phone 703-998-0660, fax 703-998-0667, jdahmann@dmso.mil

Jeffrey Olszewski
Science Applications International Corporation, 1100 Glebe Rd, Ste 1010, Arlington, VA 22201, phone 703-907-2500, fax 703-312-5940, jolszewski@std.saic.com

Richard Briggs
Virtual Technology Corporation, 5458 Midship Ct, Burke, VA 22015, phone 703-978-2211, fax 703-913-2140, rbriggs@virtc.com

Russell Richardson, PhD
Science Applications International Corporation, 1100 Glebe Rd, Ste 1010, Arlington, VA 22201, phone 703-907-2500, fax 703-312-5940, rrichardson@std.saic.com

Richard M. Weatherly, PhD
The MITRE Corporation, 7525 Colshire Dr, McLean, VA 22101-3481, phone 703-883-7203, fax 703-883-3343, weather@mitre.org

James Calvin
MIT Lincoln Lab, 244 Wood St., Lexington, MA 02173, phone 617-981-4116, fax 617-981-7455, jcalvin@ll.mit.edu

Philomena Zimmerman
Naval Air Warfare Center-Aircraft Division, 48150 Shaw Rd. 5115, Bldg 2109, Unit 5, NAS Patuxent River, MD 20670-5304, phone 301-342-6362, fax 301-342-6381, zimmermanp%am7@mr.nawcad.NAVY.MIL

Keywords: HLA, RTI, Performance, Benchmark, DMSO

ABSTRACT: *This paper describes DMSO efforts on HLA performance, and presents information representing the HLA performance framework development process underway by the DMSO and the Department of Defense Architecture Management Group (AMG). The AMG is establishing a performance framework for HLA . The performance framework is a standard method for the characterization of HLA federations and the measurement of the performance of components of a federation, including the RTI. Benchmarking an RTI using this framework will permit comparisons to be made between different RTI implementations. Within the simulation community, different aspects of RTI performance are of interest. The AMG is attempting to address these diverse needs in its design of the performance framework. This paper describes the progress in this area to date.*

1. Introduction

The US Department of Defense has developed the High Level Architecture to support the reuse and interoperability of simulations [1,2,3,4]. The HLA applies across a broad range of simulation applications including “as fast as possible” closed form analyses, human-in-the-loop training systems, and hard real-time hardware-in-the-loop test and evaluation applications.

In HLA applications, ‘federations’ of simulations are formed by bringing together simulations and support tools which as a group offer the set of representations and other capabilities (e.g., data collectors, interfaces to live systems, management tools) needed to meet the needs of a particular user [5]. At runtime, the simulations and other federates exchange necessary data using those services of the runtime infrastructure (RTI) which meet the needs of the federation.

The HLA, and the interfaces to the RTI, are designed to provide a federation a set of services which can be used as needed to meet the objectives of the federation. Different federations will use different services and will have different performance requirements for the operations of a federation as a whole and for the different components of the federation. In particular, different federations will require the RTI to provide services (delivery of data, progression of logical time) within certain tolerances for the overall operation of the federation to meet their needs. Similarly, each federation will also have performance requirements for other components of the federation, including participating federates, since a running federation operates as a single, distributed system.

1.1 Why a performance framework?

As HLA is adopted and used for a variety of applications, it is becoming increasingly important that processes and tools are developed which support the user in employing HLA to meet their needs. A number of efforts are underway [5,6,7] which address the general process for developing and executing HLA federations, an architecture for development of HLA supporting tools, and standards to support common HLA object model data contents to promote reuse using the HLA.

Beyond this, methods and, eventually, tools are needed to help HLA users configure federations with components which have the capacity and performance to meet the needs of their applications. Towards this end, efforts have been initiated to develop an HLA performance framework. Initially motivated by the need to understand RTI performance requirements and capabilities, this effort is providing a foundation for broader use supporting planning of federation executions and defining expected performance of other components of HLA federations including federate capabilities, hardware and network requirements

1.2 Federation Execution (Fedex) Planners Workbook Development

Under the auspices of the DoD Architecture Management Group (AMG) a series of technical exchanges have been held to define the aspects of federations which bear on their runtime performance. After several iterations, the Federation Execution (Fedex) Planners Workbook, described in this paper, reflects the current status of these efforts. The workbook is comprised of a series of tables which call for information about a federation execution which has been deemed relevant to the federation runtime performance. The workbook has been developed based on inputs from HLA users and from RTI developers and testers, and has been used including the JADs program, to describe specific HLA federation executions or execution plans for a series of HLA applications and RTI testing activities. The experience of the workbook users has been factored into the workbook development which is a continuing activity.

1.3 Fedex Planners Workbook

The Federation Execution Planning Workbook is made up of a set of five tables:

- Federation Execution Summary Table
- Host Table
- LAN Tables
- RTI Services Tables
- Object/Interaction Table

One completed workbook describes a federation execution (fedex). While the workbook requests information about a range of aspects of the fedex to provide the key dimensions for assessing performance of the fedex, these are all aspects that a fedex developer would necessarily need to understand in order to operate a fedex. In this light, this workbook provides a first step in definition of the Federation Required Execution Details (FRED) as called out in the Federation Execution Development and Execution Process (FEDEP) [5].

2. Description of Federation Execution Planning Workbook Tables

Each of the tables in the workbook is described briefly below and are shown in a series of figures which are presented at the end of this paper. As the workbook is used with different cases it will be revised based on this experience.

2.1 Federation Execution Summary Table (Figure 1)

One Federation Execution Summary Table is completed for a fedex. This table includes information about the fedex itself (name, number of concurrent fedexes) and summary information about each federate in the fedex (API used, settings of time management switches). It also identifies hosts and LANs for the federates; these are described in the following tables

2.2 Host Table (Figure 2)

The host table requests information on the hardware, software and capacity of the hosts which support the federates in the fedex. One host table is completed for a fedex.

2.3 LAN Tables (Figure 3)

The LAN tables requests descriptive information on each LAN used in the fedex and the LAN-to-LAN connections. One set of LAN tables is complete for a fedex.

2.4 RTI Services Tables (Figure 4)

This table lists the current suite of RTI services and requests that any service used at least once in the fedex be identified. One RTI services table is completed for a fedex.

2.5 Object/Interaction Table (Figure 5)

One Object/Interaction table is completed *for each federate* in a fedex. This table identifies which attributes of object are updated by each federate, how often and in what groupings. Likewise the table identifies those attributes to which a federate subscribes and latency constraints on updates. Similar data is recorded on this table for interactions.

3. Uses of the Workbook

The Fedex Planners Workbook can be used by the HLA community for a number of different purposes.

First it is already assisting developers of fedex's to configure the various components of a fedex to meet the needs of their application. For HLA users in the early stages of use of HLA, it provides insights into the latter stages of the federation development and execution process (FEDEP) [2,3,4]. In fact, this workbook has become the first incarnation of a capability called out in the FEDEP as the "federation required execution detail" or (FRED).

Second, the workbook provides a common mechanism to collect data from a range of HLA users to understand the ways different HLA user communities are applying HLA and the performance needs, particularly performance of RTI services, of these communities. It is envisioned that once a sufficient pool of examples has been collected that a set of nominal or benchmark federation executions might be defined that typify the range of applications anticipated. These could be used to direct, or to evaluate the suitability of, support software such as the RTI.

Third, the workbook is already being employed by RTI testers to describe the federation execution context for RTI testing and work is underway to define the RTI-specific performance metrics and tools which can be used with different RTI implementations to benchmark their performance. These metrics and supporting tools will quantify RTI performance in terms of selected sets of RTI functionality, including measurements of end-to-end latency, RTI throughput, ownership exchange rate and time synchronization rate. Data from the Fedex planners workbook will be used to derive performance expectations for RTI services which can be measured using these metrics and supporting tools.

Fourth, the workbook is useful to federate developers in defining and profiling the performance characteristics of their federates. As HLA matures, increasingly users will seek simulations to incorporate into HLA federations to meet their needs. Efforts are underway to support this through the development of an Object Model Library which will allow federation developers to identify candidate participants in their federations based on the types of data simulations are able to offer to a federation. Beyond this however, use of a federate in a particular federation will require certain performance on the part of the federate. The workbook provides a starting point for defining these federate performance characteristics, and eventually metrics and tools like those underway for the RTI.

Finally, the workbook provides a framework for development of automated tools to support fedex planning and testing. As with other parts of HLA (e.g., object model development), the configuring and testing of possible configurations is an excellent candidate for automated support. The workbook identifies many of the aspects of a federation which would be incorporated into tools supporting this aspect of federation development.

4. A Sample Use of the Framework

A sample use of the framework is depicted in Figures 1-5, as drawn from the test configuration described in Figure 6.

5. Summary

In summary, processes, metrics and tools are needed to support user design of HLA federations which meet the performance requirements of applications. The Fedex Planners Workbook is a first step in this direction. Developed with input from a range of HLA federation developers as well as RTI developers and testers, it provides a structured way to describe the information about a particular federation execution relevant to its performance. It assists a fedex planner in the configuration of their fedex, and provides the information needed to assess particular performance needs of that federation in terms of the federates, RTI services and hardware and network capacities. Beyond this, it provides a common frame of reference for future developments of common tools and metrics for HLA system design and operations

6. References

[1] J. Dahmann: “High Level Architecture for Simulation”, Proceedings of the IEEE Conference , Israel, March 1997.
[2] “High Level Architecture Interface Specification, V1.1” Defense Modeling and Simulation Office, 12 February 1997 (http://hla.dmso.mil).
[3] “High Level Architecture Object Model Template, V1.1” Defense Modeling and Simulation Office, 12 February 1997 (http://hla.dmso.mil).
[4] “High Level Architecture Rules, V1.0” Defense Modeling and Simulation Office, 15 August 1996 (http://hla.dmso.mil).
[5] J. Dahmann, D. Ponikvar, R. Lutz: “HLA Federation Development and Execution Process”, Proceedings of the Distributed Interactive Simulation Workshop, Orlando, FL, September, 1996
[6] K. H, J. Dahmann, R. Lutz, J. Sheehan: “Planning For the Evolution of Automated Tools in HLA”, Proceedings of the 1997 Spring Simulation Interoperability Workshop, Orlando, FL, March, 1997
[7] R. Scrudder, J. Sheehan: “HLA FOM/SOM Content Standards”, Proceedings of the 1997 Spring Simulation Interoperability Workshop, Orlando, FL, March, 1997

Authors Biography

JUDITH DAHMANN was named as DMSO's chief scientist in July, 1995. Prior to this, Dr. Dahmann worked as a Program Area Manager for the MITRE Corporation in the area of advanced simulation programs. Dr. Dahmann has over 8 years of experience in defense modeling and simulation and her most notable accomplishments include development of the Aggregate Level Simulation Protocol (ALSP) architecture for distributed simulation. Dr. Dahmann holds a Masters degree from the University of Chicago, and a Doctoral degree from the Johns Hopkins University.

RICHARD WEATHERLY is the Chief Engineer for the MITRE Corporation in their C3 Modeling and Simulation Center. He received his Ph.D. in 1984 from Clemson University in Electrical Engineering. Currently, he is the project leader for the RTI F.0, 1.0, and 1.1 developments. Please see http://ms.ie.org/weatherly/ for publications and professional information.

RUSS RICHARDSON is a Program Manager and Assistant Vice President of Simulation Technology for Science Applications International Corporation. He received his BS, MS, and PhD degrees in electrical engineering from the Ohio State University. He is the technical director of the Joint Precision Strike Demonstration (JPSD) ACTD Program. He has extensive background in all areas related to RTI development. Currently, he is responsible for the design and development of the Integration and Evaluation Center (IEC) located at the U.S. Army Topographic Engineering Center, Ft. Belvoir, Va. In this capacity he provides overall technical direction to a large (43+) interdisciplinary, multi-contractor team creating an engineering testbed. The focus of the testbed is the integration of Distributed Interactive Simulation (DIS) and network technology with live tactical systems.

PHILOMENA ZIMMERMAN is the Distributed Simulation Project Lead for the Warfare Simulation Branch of Air Combat Environment Test and Evaluation Facility (ACETEF) at the Naval Air Warfare Center - Aircraft Division (NAWC-AD). She is currently involved in the HLA as a member of the Engineering Federation, a member of the HLA TSTCore, and lead of the HLA Security IPT. She is also the technical lead for the Warfare Simulation Branch in the JADS-EW Test. Ms. Zimmerman has been involved in the interoperability standards effort since 1990, and is currently a member of the PRP for the Federation Development Process Forum. She received her B.S. in Mathematics from St. John Fisher College.

JAMES CALVIN is Assistant Group Leader in the Distributed Simulation Systems Group at MIT, Lincoln Laboratory. He received his M.S. in Engineering Design from Tufts University. He has worked on the STOW RTI-S design and development, and is now focusing on RTI 1.1 design, development, and integration.

RICHARD BRIGGS is a Director and System Engineer for Virtual Technology Corporation. He received his BS in Computer Science from Pennsylvania State University. He is the technical lead for the RTI F.0 support team helping DMSO establish a support infrastructure to mitigate users' risks in migrating to the HLA. He was actively involved in the HLA prototyping efforts and is currently working with the HLA management interface working group and the DIS++ protocols working group. For the HLA prototyping efforts he was lead engineer for the JPSD Experiment (an HLA prototype effort) and the lead software architect of an extensible object-oriented framework used to integrate DIS simulations and C4I systems with the RTI prototype.

JEFFREY OLSZEWSKI is a computer scientist in the Simulation Technology Division of SAIC, and is currently the technical lead for the HLA Testbed. He received his B.Sc. degree in computer science from the University of Pittsburgh, and will complete his M.Sc. in Computer Science from George Washington University in 1997.

Federation Execution Summary Table

Federation Execution Name	Jager_test
Number of Concurrent Federation Executions (total including this Federation	**1**
(If more than one, list names of others beyond this	
RTI Software Used (Version)	1

Federate Summary

	Name	API (C++,Ada, IDL,	Tic Rate	Time Management Regulating (y or	Constraining (y or	Host (assign # to each [List data on Host	LAN (assign # to LAN) [List data on LAN
Fed $_1$	JAGER_1	C++	14/sec	N	N	1	1
Fed $_2$	JAGER_2	C++	14/sec	N	N	2	1
Fed $_3$	JAGER_3	C++	14/sec	N	N	3	1
Fed $_4$	JAGER_4	C++	14/sec	N	N	4	1
Fed $_5$							
Fed $_6$							

Figure 1. Federation Execution Summary Table

Host Table

	Hardware	Operating System	Memory available to RTI (MB)	Total CPU Available to Federation and RTI Combined	% CPU Available to RTI
$Host_1$	Sun Ultra2, 2x200MHZ ULtraSPARC, 256M ram	Solaris 2.5.5	54%, of 256 Meg measured	95% CPU available, nominal load	25%
$Host_2$	Sun Ultra2, 2x200MHZ ULtraSPARC, 256M ram	Solaris 2.5.5	54%, of 256 Meg measured	97% CPU available, nominal load	25%
$Host_3$	Sun Ultra1, 168MHZ ULtraSPARC, 256M ram	Solaris 2.5.5	67%, of 256 Meg measured	97% CPU available, nominal load	25%
$Host_4$	Sun Ultra1, 168MHZ ULtraSPARC, 256M ram	Solaris 2.5.5	67%, of 256 Meg measured	97% CPU available, nominal load	25%

Figure 2. Host Table

LAN Tables

NOTE:
Complete one of these tables for each Federation execution

LAN Table 1: LAN Descriptions

	Physical Type (Ethernet, ATM,etc.)	**Throughtput Available to FEDEX**
LAN_1	Ethernet	Clean LAN 10Mbits/sec *80%

LAN Table 2: LAN to LAN Connectivity

	LAN_1
LAN_1	1. N/A 2. N/A 3. N/A

*1. Device type means type of switch employed to connect the LANs
*2. Throughput means the effective throughput available through the LAN connection for Federation execution, expressed in Mb
*3. Latency contribution from devices connecting the LANs, as well as the WAN, expressed in milliseconds.

Figure 3. LAN Table

RTI Services Table (Check if service to be used at least once during this Federation execution)		
Service	**IF Spec v1.1 Ref**	**Service Used?**
Create Federation Execution	2.1	x
Destroy Federation Execution	2.2	x
Join Federation Execution	2.3	x
Resign Federation Execution	2.4	x
Request **Pause**	2.5	
Initiate **Pause**	2.6	
Pause Achieved	2.7	
Request **Resume**	2.8	
Initiate **Resume**	2.9	
Resume Achieved	2.10	
Request **Federation Save**	2.11	
Initiate **Federation Save**	2.12	
Federation Save Begun	2.13	
Federation Save Achieved	2.14	
Request **Restore**	2.15	
Initiate **Restore**	2.16	
Restore Achieved	2.17	
Publish Object Class	3.1	x
Subscribe Object Class **Attributes**	3.2	x
Publish Interaction	3.3	x
Subscribe Interaction	3.4	x
Control Updates	3.5	x
Control Interactions	3.6	x
Request ID	4.1	x
Register Object	4.2	x
Discover Object	4.3	x
Update Attribute Values	4.4	x
Reflect Attribute Values	4.5	x
Send Interaction	4.6	x
Receive Interaction	4.7	x
Delete Object	4.8	x
Remove Object	4.9	x
Change Attribute Transportation Type	4.10	
Change Attribute Order Type	4.11	
Change Interaction Transportation Type	4.12	
Change Interaction Order Type	4.13	
Request Attribute Value Update	4.14	
Provide Attribute Value Update	4.15	
Retract	4.16	
Reflect Retract	4.17	
Request Attribute **Ownership Divestiture**	5.1	
Request Attribute **Ownership Assumptio**	5.2	
Attribute Ownership **Divestiture Notificati**	5.3	
Attribute Ownership **Acquisition Notifica**	5.4	
Request Attribute **Ownership Acquisition**	5.5	
Request Attribute **Ownership Release**	5.6	
Query Attribute Ownership	5.7	
Inform Attribute Ownership	5.8	
Is **Attribute Owned by Federate?**	5.9	
Request **Federation Time**	6.1	x
Request **LBTS**	6.2	
Request **Federate Time**	6.3	
Request **Min Next Event Time**	6.4	
Set **Lookahead**	6.5	
Request **Lookahead**	6.6	
Time Advance Request	6.7	x
Next Event Request	6.8	
Flush Queue Request	6.9	
Time Advance Grant	6.10	x

NOTE: Complete one of these tables for each Federation execution

Figure 4. RTI Services Table

Object/ Interaction Table

Federate #1
JAGER 1

					If Update =" y"							If Subscribe = y?
Object/Interaction Class	Attribute	Count	Size	Update? Send? (y or n)	Update Rate # updates/unit time Nominal	Maximum	Update Conditions* *Not required if FOM provided	Update Grouping (Assign same letter to attributes which will all be updated at the same time)	Transport R= Reliable B= Best Effort	Ordering TSO or FIFO	Subscribe? (y or n)	Maximum tolerable latency from any source (milliseconds)
Obj $_1$		10										
	Durability		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Damage		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Location		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Mass		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Orientation		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Shape		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Velocity		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
Interaction $_1$												
	Collide		8 bytes	y	0.3/sec	2/sec	ships collide				y	200
	Fire		8 bytes	y	0.3/sec	2/sec	user input				y	200

Denotes an invalid cell for entry of data

Object/ Interaction Table

Federate #1
JAGER 2

					If Update =" y"							If Subscribe = y?
Object/Interaction Class	Attribute	Count	Size	Update? Send? (y or n)	Update Rate # updates/unit time Nominal	Maximum	Update Conditions* *Not required if FOM provided	Update Grouping (Assign same letter to attributes which will all be updated at the same time)	Transport R= Reliable B= Best Effort	Ordering TSO or FIFO	Subscribe? (y or n)	Maximum tolerable latency from any source (milliseconds)
Obj $_1$		10										
	Durability		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Damage		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Location		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Mass		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Orientation		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Shape		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Velocity		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
Interaction $_1$												
	Collide		8 bytes	y	0.3/sec	2/sec	ships collide				y	200
	Fire		8 bytes	y	0.3/sec	2/sec	user input				y	200

Denotes an invalid cell for entry of data

Object/ Interaction Table

Federate #1
JAGER 3

					If Update =" y"							If Subscribe = y?
Object/Interaction Class	Attribute	Count	Size	Update? Send? (y or n)	Update Rate # updates/unit time Nominal	Maximum	Update Conditions* *Not required if FOM provided	Update Grouping (Assign same letter to attributes which will all be updated at the same time)	Transport R= Reliable B= Best Effort	Ordering TSO or FIFO	Subscribe? (y or n)	Maximum tolerable latency from any source (milliseconds)
Obj $_1$		10										
	Durability		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Damage		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Location		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Mass		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Orientation		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Shape		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Velocity		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
Interaction $_1$												
	Collide		8 bytes	y	0.3/sec	2/sec	ships collide				y	200
	Fire		8 bytes	y	0.3/sec	2/sec	user input				y	200

Denotes an invalid cell for entry of data

Object/ Interaction Table

Federate #1
JAGER 4

					If Update =" y"							If Subscribe = y?
Object/Interaction Class	Attribute	Count	Size	Update? Send? (y or n)	Update Rate # updates/unit time Nominal	Maximum	Update Conditions* *Not required if FOM provided	Update Grouping (Assign same letter to attributes which will all be updated at the same time)	Transport R= Reliable B= Best Effort	Ordering TSO or FIFO	Subscribe? (y or n)	Maximum tolerable latency from any source (milliseconds)
Obj $_1$		10										
	Durability		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Damage		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Location		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Mass		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Orientation		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Shape		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
	Velocity		8 bytes	y	14/sec	14/sec	eriod fixed rat	A	R	FIFO	y	200
Interaction $_1$												
	Collide		8 bytes	y	0.3/sec	2/sec	ships collide				y	200
	Fire		8 bytes	y	0.3/sec	2/sec	user input				y	200

Denotes an invalid cell for entry of data

Figure 5. Object/Interaction Table

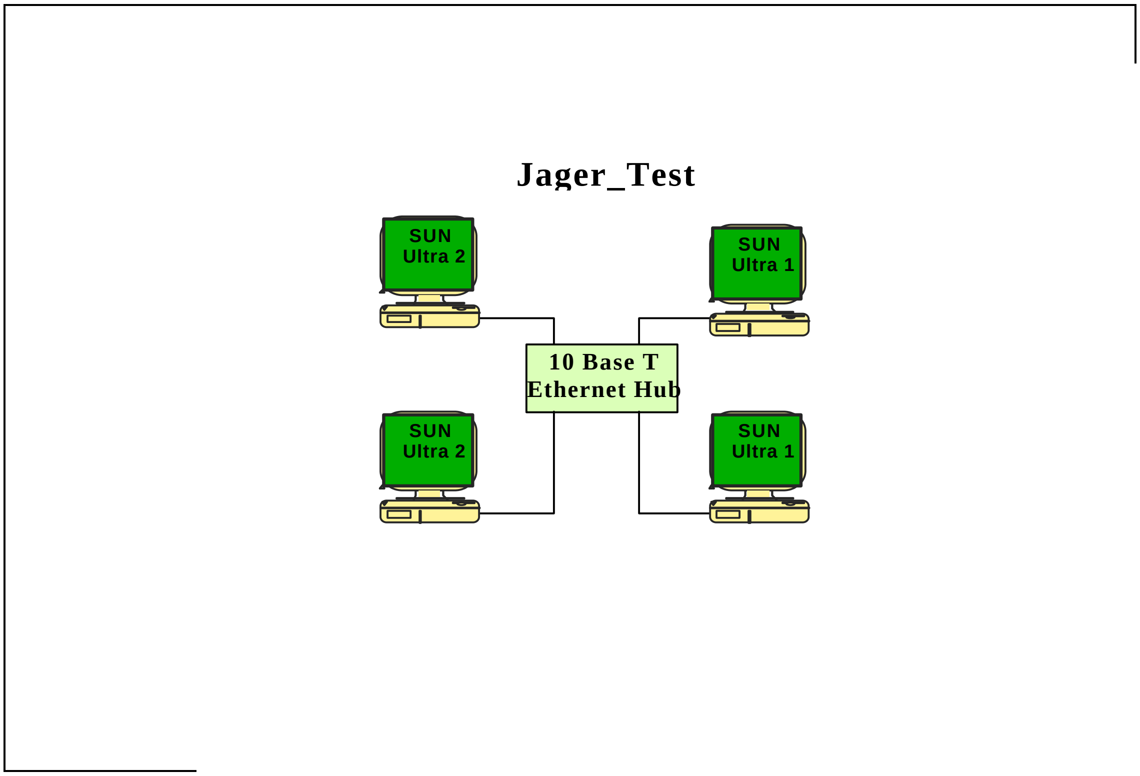

Figure 6. Jager Test Configuration